Optimization-based Uncertainty Attribution Via Learning Informative Perturbations

Hanjing Wang[1], Bashirul Azam Biswas[1], and Qiang Ji[1]

Rensselaer Polytechnic Institute, Troy NY 12180, USA
{wangh36, biswab, jiq}@rpi.edu

Abstract. Uncertainty attribution (UA) aims to identify key contributors to predictive uncertainty in deep learning models. To improve the faithfulness of existing UA methods, we formulate UA as an optimization problem to learn a binary mask on the input. The learned mask identifies regions that significantly contribute to output uncertainty and allows uncertainty reduction through learning informative perturbations on the masked input. Our method enhances UA interpretability and maintains high efficiency by integrating three key improvements: Segment Anything Model (SAM)-guided mask parameterization for efficient and interpretable mask learning; learnable perturbations that adaptively target and refine problematic regions specific to each input without manually tuning the perturbation parameters; and a novel application of Gumbel-sigmoid reparameterization for efficiently learning Bernoulli-distributed binary masks under continuous optimization. Our experiments on problematic region detection and faithfulness tests demonstrate our method's superiority over state-of-the-art UA methods.

Keywords: Uncertainty Attribution · Model Explainability

1 Introduction

While conventional deep learning has made remarkable strides in various domains, they often face challenges with overconfidence in their predictions, which can be risky in safety-critical applications. Hence, uncertainty quantification (UQ) is paramount, as it directly impacts the safety, reliability, and trustworthiness of AI systems, especially for various computer vision applications [8, 15, 48]. Predictive uncertainties in deep learning are categorized into aleatoric uncertainty, reflecting inherent data randomness, and epistemic uncertainty, resulting from limited model knowledge. Bayesian Neural Networks (BNNs) serve as a foundational approach for quantifying uncertainty, encompassing techniques such as Variational Inference (VI) [7, 19, 22] and Markov Chain Monte Carlo (MCMC) [36, 42, 47] methods to estimate the complex posterior distribution of model parameters. Following this, Ensemble methods [16, 37, 38, 43] offer an alternative strategy, aggregating predictions from multiple models to capture uncertainty. While numerous studies have focused on enhancing the efficiency and accuracy of UQ, comparatively little effort has been devoted to understanding the sources of uncertainty. Identifying the sources of uncertainty is essential,

as it reveals where and why certain inputs are associated with high uncertainty for enhanced explainability, thereby informing more targeted improvements in model design and training.

Uncertainty attribution (UA) [1, 26, 39, 40] systematically identifies and analyzes the specific inputs or features that contribute significantly to the uncertainty in the model's predictions. Traditional explainable AI methods highlight areas of inputs leading to specific predictions, which are unreliable when applied directly to UA for identifying problematic regions in input data. The primary reason is that they focus on explaining the model's predictions, assuming confident predictions, while UA aims to pinpoint the sources of input imperfections that contribute to high predictive uncertainties. Additionally, these methods are often class-discriminative when applied to explain the classification score. Consequently, they may struggle to explain inputs with incorrect predictions and large uncertainties [26]. Uncertainty attribution methods are broadly categorized into two types: gradient-based and perturbation-based methods. Gradient-based methods leverage the gradients of the model output with respect to the input to identify the areas contributing to uncertainty. On the other hand, perturbation-based methods involve systematically altering parts of the input and observing the effect on uncertainty, allowing for a more direct assessment of how variations in the input influence the uncertainty. While gradient-based UA methods are often efficient, they only consider the uncertainty sensitivity to very small input changes, leading to noisy and sometimes misleading indications of uncertainty. Perturbation-based methods can be computationally intensive and may be less accurate if an insufficient number of perturbations are performed.

Inspired by existing literature [3, 5, 45], we treat UA as an optimization task aimed at learning a binary mask to identify problematic regions of the input. This mask targets specific areas for maximum uncertainty reduction, achieved by applying learnable perturbations to these selected areas. This optimization-based UA framework can potentially improve the faithfulness of UA, which quantifies how accurately a method's explanations reflect the real impact of input features on the uncertainty. A method is considered faithful if the identified areas truly correspond to regions in the input that, when perturbed, have a significant impact on the predictive uncertainty. Our approach ensures faithfulness through a direct linkage between the uncertainty and the identification of uncertain regions in the optimization objective. Moreover, compared to perturbation-based methods, our approach utilizes gradients to iteratively update the mask through continuous optimization, avoiding the need for computationally intensive, pixel-wise input perturbations. This strategy may reduce the cost associated with exploring an exponential number of pixel combinations. Unlike gradient-based methods, which directly employ gradients to generate uncertainty maps, our method applies actual perturbations to inputs, not just minor input changes, potentially yielding more accurate explanations.

To enhance UA performance, we introduce three key adjustments. First, we use a linear combination of segments from the pre-trained Segment Anything Model (SAM) [12] to parameterize the mask. Using SAM segments as a basis

enables targeted analysis of semantic regional effects and simplifies the learning process to linear combination coefficients instead of pixel-level masks. Second, traditional perturbations used to explain model predictions often fall short in UA. For example, widely used mask-out perturbations invariably result in increased uncertainty in UA, even when applied to problematic regions. This underscores the need for advanced, learnable perturbations in UA, which are essential for accurately reducing uncertainty by modifying problematic regions. Finally, given the challenge of learning a binary mask directly, we adopt a Bernoulli distribution for the masks and employ the Gumbel-sigmoid reparameterization trick [10], enabling continuous optimization with high efficiency.

2 Related Work

Gradient-based Attribution Methods. Gradient-based methods [11, 29, 31–35, 44, 46] leverage gradient information as input attribution. However, raw gradients can be noisy, leading researchers to develop various approaches for refining gradient information. SmoothGrad [32] addresses the issue of noisy gradients by aggregating attributions from multiple noisy images. FullGrad [34] combines the gradients of the output with respect to both the input and the bias variable in each convolutional or fully-connected layer. Integrated Gradients (IG) [35] and its variants [11, 44] create a path integral between a reference image and the input. They then aggregate the gradients of the model output for the artificial images along this path, providing a more comprehensive understanding of the input-output relationship. Grad-CAM [29] focuses on the gradients of the latent feature maps, offering more interpretable insights. Layer-wise relevance propagation (LRP) methods [2, 23, 30] construct attributions by backpropagating the relevance score layer-by-layer from the output to the input. While most gradient-based methods can be applied for both model prediction and uncertainty attribution, LRP methods often need specific neural network architectures [39]. The direct application of these attribution methods might not be sufficient for UA, leading Wang et al. [39] to develop a backpropagation method specifically for UA. In summary, gradient-based methods are often computationally efficient to analyze neural network input-output relationships. Yet, they mainly model output changes from very small input perturbations, which may not yield precise UA results. This limitation is not effectively overcome by refining gradient information through smoothing, aggregation, and path integration techniques.

Perturbation-based Attribution Methods. Perturbation-based methods [3, 5, 6, 27, 45, 46] estimate attributions by modifying the input and observing the subsequent changes in the model's output. For example, LIME [28] approximates the model's behavior locally by fitting an interpretable model to the perturbed data. RISE [27] generates saliency maps by randomly masking the input and aggregating the model's responses. SHAP and its variants [20, 21] leverage Shapley values from cooperative game theory to interpret complex model predictions, assigning importance to each feature by observing output changes upon feature

perturbations. The SHAP-based methods are extended to UA with theoretical studies [41]. While perturbation-based methods provide a more intuitive understanding of the model's behavior and allow for better interpretability, they can also be computationally expensive. This is primarily due to the need for an exponentially increasing number of perturbations and subsequent model evaluations to explore the input-output relationship effectively. To improve efficiency, Dabkowski et al. [3], Fong et al. [5], and Yang et al. [45] focus on identifying the smallest regions for perturbation that can significantly change the model's prediction, leveraging continuous optimization techniques to speed up the attribution process. In recent years, several methods have been specifically designed for UA. For instance, CLUE [1] and its variants [17,18] focus on creating an improved image with minimal uncertainty by modifying the uncertain input using a generative model. The attribution map is then generated by measuring the difference between the original and modified input. To further enhance pixel-wise attributions, Perez et al. [26] combined the CLUE method with the path integral technique. However, CLUE-based methods require a pre-trained generative model, which may be challenging to acquire for practical applications.

3 Proposed Method

3.1 Preliminary

Bayesian Neural Networks. Distinct from point-estimated neural networks, BNNs treat neural network parameters as random variables and calculate the posterior distribution of the parameters $\boldsymbol{\theta}$, denoted as $p(\boldsymbol{\theta} \mid \mathcal{D})$ where $\mathcal{D}$ represents the training data. Utilizing Bayes' rule, the posterior distribution can be computed using $p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$ with $p(\mathcal{D}|\boldsymbol{\theta})$ as the likelihood and $p(\boldsymbol{\theta})$ as the prior. Since the exact posterior distribution is often intractable, various techniques have been developed, including MCMC sampling, VI and Ensemble methods. These approaches either directly sample from the posterior or approximate it using a simpler distribution for efficient sampling.

Uncertainty Quantification. In this study, we concentrate on image classification tasks, where $\mathbf{x}$ denotes the input and $\mathbf{y}$ denotes the output. The neural network predicts $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$, which is a categorical distribution parameterized by learnable softmax probability. Entropy is a common method for quantifying the uncertainty or randomness in a probability distribution. In entropy-based UQ, the total uncertainty of predictions is decomposed as follows [4]:

$$\underbrace{\mathcal{H}\left[p(\mathbf{y}|\mathbf{x}, \mathcal{D})\right]}_{\text{Total Uncertainty } U_t(\mathbf{x})} = \underbrace{\mathcal{I}\left[\mathbf{y}, \boldsymbol{\theta}|\mathbf{x}, \mathcal{D}\right]}_{\text{Epistemic Uncertainty } U_e(\mathbf{x})} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})]\right]}_{\text{Aleatoric Uncertainty } U_a(\mathbf{x})} \quad (1)$$

Here, $\mathcal{H}$, $\mathcal{I}$, and $\mathbb{E}$ stand for entropy, mutual information, and expectation, respectively. By employing a Monte Carlo approximation of the posterior, we can

approximate the uncertainties as:

$$\mathcal{H}\left[p(\mathbf{y}|\mathbf{x},\mathcal{D})\right] = \mathcal{H}\left[\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})]\right] \approx \mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S} p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}^s)\right] \tag{2a}$$

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})]\right] \approx \frac{1}{S}\sum_{s=1}^{S} \mathcal{H}[p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}^s)] \tag{2b}$$

where $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|\mathcal{D})$ represents the sampling of parameters $\boldsymbol{\theta}$ from their posterior distribution, and S is the sample size.

3.2 Optimization-based Uncertainty Attribution

Our approach defines UA as an optimization problem, aiming to learn a binary mask, $\mathbf{M}$, with the same dimensions as $\mathbf{x}$, that highlights areas in the inputs significantly contributing to uncertainty. For each element u in $\mathbf{x}$, we assume $\mathbf{M}(u) \in \{0, 1\}$, with $\mathbf{M}(u) = 1$ indicating strong contributions to predictive uncertainty and $\mathbf{M}(u) = 0$ otherwise. To estimate each element's contribution, we perturb the regions indicated by $\mathbf{M}$. Specifically, with $\hat{\mathbf{x}}$ representing the perturbed input with reduced uncertainty, we replace the regions where $\mathbf{M}(u) = 1$ in $\mathbf{x}$ with those from $\hat{\mathbf{x}}$, resulting in $\mathbf{x}^{'} = (1 - \mathbf{M}) \odot \mathbf{x} + \mathbf{M} \odot \hat{\mathbf{x}}$ to denote the masked input. The goal is to identify a small subset of $\mathbf{x}$'s regions, $\mathbf{M}$, that significantly reduce predictive uncertainty through the masked input. The uncertainty, denoted as U, could be aleatoric (U_a), epistemic (U_e), or total (U_t), as indicated in Eq. (1). The optimal $\mathbf{M}^*$ is determined by:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} U((1 - \mathbf{M}) \odot \mathbf{x} + \mathbf{M} \odot \hat{\mathbf{x}}) + \lambda||\mathbf{M}|| \tag{3}$$

Here, $\odot$ denotes element-wise multiplication, $||\mathbf{M}||$ is the norm of $\mathbf{M}$, and λ is a hyperparameter balancing uncertainty reduction and mask sparsity, usually prioritizing uncertainty loss over sparsity regularization. As U utilizes all ensemble models sampled from $p(\boldsymbol{\theta}|\mathcal{D})$ to compute uncertainty, the UA maps are learned by jointly backpropagating through these models. Perturbations in identified problematic regions decrease predictive uncertainty, while those in high-certain areas will increase uncertainty, guiding $\mathbf{M}$ to select critical regions for perturbation to reduce uncertainty. Background or non-informative regions will not impact uncertainty after perturbation, necessitating the inclusion of a sparsity term to ensure these areas are assigned zero values in $\mathbf{M}$.

3.3 SAM-guided Mask Parameterization

To enhance the explainability of our proposed method and reduce the optimization complexity, we aim to parameterize the uncertainty mask, $\mathbf{M}$, through a linear combination of segments derived from the pre-trained SAM model, denoted as $\{\mathbf{M}_i\}_{i=1}^{N}$. These N segments are binary masks of the same dimension as $\mathbf{M}$, each highlighting distinct regions of the input space. Generated by the

SAM model, these segments are inherently binary and delineate areas corresponding to semantically meaningful and human-understandable concepts. The parameterization of $\mathbf{M}$ is $\mathbf{M} = \sum_{i=1}^{N} w_i \mathbf{M}_i$ where $w_i \in \{0, 1\}$ represents the binary weights. By employing these segments as the basis for constructing $\mathbf{M}$, our method not only identifies regions contributing to uncertainty but also ensures that these regions are aligned with recognizable semantic features. This alignment significantly boosts the explainability of our method, as the identified areas of uncertainty are directly tied to concepts that are easily interpretable by humans. For instance, in an image classification task, rather than indicating abstract or non-intuitive areas of uncertainty, our method could highlight specific objects or parts of an image that contribute to the uncertainty, such as "eyes" in a facial recognition task or "vehicles" in an autonomous driving context.

While the effectiveness of our methods depends on the accuracy of SAM, we believe our approach can still be effective with imperfect SAM segmentation. For instance, if problematic regions are either included in a large SAM segment or spread across multiple segments, our method can sometimes still identify these areas as potentially problematic. Furthermore, our method is compatible with state-of-the-art segmentation models, ensuring adaptability and improved performance across diverse applications with domain-specific segmentation models.

3.4 Learnable Perturbation

Traditional perturbation-based methods often use fixed or random perturbations, which may not effectively reduce uncertainty and could unintentionally increase it, even in problematic regions. Therefore, we dynamically adjust learnable perturbations during the optimization process, specifically targeting and modifying areas in the input $\mathbf{x}$ to effectively reduce uncertainty.

Our approach updates the perturbed input $\hat{\mathbf{x}}$, as described in Eq. (3), through iterative blurring, represented as $\hat{\mathbf{x}} = g_\phi(\mathbf{x})$, where $g_\phi(\cdot)$ is a blurring function parameterized by ϕ. This strategy hypothesizes that blurring problematic regions decreases uncertainty, while blurring good areas could increase it by obscuring important information for prediction. Recognizing that a universal blurring strategy does not cater to the varied demands of different inputs, we aim to learn the blurring function's parameters ϕ within the optimization process. This approach allows for the precise, dynamic adjustment of perturbations, blurring each region as needed to optimally reduce uncertainty.

3.5 Gumbel-sigmoid Reparameterization For Binary Weights

To address the challenge of modeling binary weights for our uncertainty mask, $\mathbf{M}$, we confront the complexities of discrete optimization, which demands extensive search efforts and complicates direct learning of these weights. While continuous optimization through iterative gradient descent offers a potential solution, simply relaxing binary weights to a continuous range between $[0, 1]$ undermines the mask's binary nature, which is crucial for pinpointing problematic areas and facilitating accurate input region replacement through perturbations.

On one hand, adhering strictly to binary weights ensures clarity in identifying areas contributing to uncertainty but, on the other hand, it restricts our ability to convey the nuanced relative importance of these regions.

In this paper, we propose maintaining the binary conception of weights while adopting a probabilistic approach for learning them. Specifically, we model each weight as an independent Bernoulli distribution, denoted as $w_i \sim \text{Ber}(p_i)$ with probability parameter $p_i \in [0, 1]$. This strategy enables the use of continuous optimization techniques to learn the probabilities, thereby circumventing the challenges associated with direct discrete optimization. Consequently, our loss function is designed to account for the expected loss over the distribution of weights. At each step of gradient descent, we sample from this distribution, using the Gumbel-sigmoid function [10, 25] as a smooth approximation to generate these samples. Let w_i^s denote the s-th sample from $\text{Ber}(p_i)$. The Gumbel-sigmoid reparameterization trick approximates the binary sample w_i^s by

$$w_i^s = \frac{1}{1 + \exp\left[-\frac{\log\left(\frac{1-p_i}{p_i}\right) + g_1^s - g_0^s}{\tau}\right]} = \sigma\left(\frac{l + t^s}{\tau}\right). \tag{4}$$

g_1^s and g_0^s are independent samples from standard Gumbel distribution, i.e., Gumbel(0,1), and τ is the temperature controlling how close w_i^s is to either 0 or 1. As shown by [25], τ is often chosen from 0.01 to 0.2 to achieve good binary approximation and optimization stability. We define the logit $l = \log(\frac{1-p_i}{p_i})$, $t^s = g_1^s - g_0^s$, and $\sigma(\cdot)$ as the sigmoid function. It can be shown that t^s is equally sampled from the standard logistic distribution (Logistic(0,1)) [25]. With a small temperature, w_i^s is close to binary. The Gumbel-sigmoid reparameterization trick facilitates the use of gradient-based optimization while preserving the binary nature of the weights. Moreover, it enhances the expressiveness of the mask by utilizing the probabilities to indicate the relative importance of each region.

3.6 Overall Process

In this paper, we incorporate a SAM-guided mask, learnable perturbation, and Gumbel-sigmoid reparameterization into Eq. (3) as follows:

$$\begin{aligned} \mathbf{p}^*, \phi^* &= \arg\min_{\mathbf{p},\phi} \mathbb{E}_{\mathbf{w}\sim\text{Ber}(\mathbf{p})} \left[U((1-\mathbf{M}) \odot \mathbf{x} + \mathbf{M} \odot \hat{\mathbf{x}}) + \lambda||\mathbf{w}||\right] \\ &\approx \arg\min_{\mathbf{p},\phi} \mathbb{E}_{\mathbf{t}\sim\text{Logistic}(0,1)} \left[U((1-\mathbf{M}) \odot \mathbf{x} + \mathbf{M} \odot \hat{\mathbf{x}}) + \lambda||\mathbf{w}||\right] \\ \text{where} \quad \mathbf{M} &= \sum_{i=1}^{N} \sigma\left(\frac{\log\frac{1-p_i}{p_i} + t}{\tau}\right) \mathbf{M}_i \quad \text{and} \quad \hat{\mathbf{x}} = g_\phi(\mathbf{x}) \end{aligned} \tag{5}$$

$\mathbb{E}$ represents the expectation, and $\mathbf{p} = [p_1, p_2, \cdots, p_n]^T$ are the probabilities for the Bernoulli distributions. $\mathbf{M}_i$ is the binary segment mask pre-generated by SAM, and $g_\phi(\cdot)$ is a blur function parameterized by ϕ. In this paper, we adopt a Gaussian blur function, and ϕ will represent the standard deviation that governs

the spread of the blur. To ensure $p_i \in [0, 1]$, we apply a sigmoid function to p_i, denoted as $p_i = \sigma(v_i)$, and learn the real number $\{v_i\}_{i=1}^N$ instead. To solve the optimization problem, we initialize $v_i = 0, \phi = 0$ for all indices i and utilize gradient descent to update them. To estimate the expectation term, we leverage the stochastic optimization method. At each gradient step, we get one sample from Logistic(0,1) to update the parameters until convergence. Compared to directly solving Eq. (3), applying a SAM-guided mask could potentially speed up the optimization process since only a small set of weight parameters need to be learned. Adding the learnable perturbation parameter ϕ will not theoretically increase complexity, as ϕ can be updated in parallel with $\{w_i\}_{i=1}^N$. If the SAM model is not available, our method can also be applied at the pixel level, treating $\mathbf{M}_i$ as a segment solely for the i-th pixel. Although pixel-wise UA is less interpretable by humans, it often contains more detailed information.

As shown in Eq. (5), the uncertainty function directly serves as the optimization objective, and the mask is learned to get the optimal uncertainty reduction, potentially yielding more accurate and faithful explanations. Compared to standard perturbation-based methods, the proposed framework avoids performing an exponential number of perturbations on the input and manually tuning the perturbation parameters. Perturbation-based methods are accurate only if the perturbations are representative. However, it is hard to randomly obtain representative perturbations for a large input space, such as high-resolution images.

4 Experiments

Dataset. We evaluate the proposed method on the benchmark image classification datasets including CIFAR-10 (C10) [14], CIFAR-100 (C100) [13], and SVHN [24]. We also evaluate our method on high-quality PartImageNet [9].
UQ Model. We adopted the deep ensemble technique [16] for quantifying uncertainty. This method involves the training of several deep neural networks, each beginning from different initialization. Notably, this technique has proven highly effective in UQ, surpassing various approximate BNNs [16]. Specifically, the UQ model for each dataset consists of 5 ensemble components, each trained independently with random initialization.We use ResNet18 and ResNet152 for C10 and C100, a two-layer CNN for SVHN, and DinoV2 with fully connected layers for PartImageNet. The implementation details of model architecture and training hyperparameters for all datasets are provided in Appendix A.1.
Baselines. We compare our proposed method with both gradient-based and perturbation-based methods. Following the experiment settings of [39], the gradient-based baselines include Grad [31], SmoothGrad [32], IG [35], and UA-Backprop [39]. The perturbation-based baselines include KernelSHAP [21], LIME [28], and Occlusion [46]. Those methods, except UA-Backprop, are general approaches for analyzing the input-output relationship and can be directly extended to UA. They have been extended to UA in [26, 39], serving as standard baselines. It is worth noting that effectively applying SAM to gradient-based attribution methods is challenging due to the dependency of uncertainty on the entire input. Yet,

SAM may fit well with some perturbation-based methods, allowing for the perturbation of segment combinations generated by SAM and avoiding the need to perturb thousands of pixel combinations. The results of the integration of KernelSHAP with SAM are also reported for the detection of problematic regions. Implementation details for all methods are shown in Appendix A.2.

Evaluation Tasks. In Sec. 4.1, we assess all methods for detecting problematic regions on semi-synthetic data, where the ground-truth problematic regions are known. In Sec. 4.2, we perform faithfulness tests on the methods to observe the reduction in uncertainty achieved by altering only the identified problematic regions. The detailed experiment settings are provided in Appendix A.3.

4.1 Detection of Problematic Regions

In this section, we aim to evaluate methods for detecting image anomalies using semi-synthetic data with known problematic areas. We introduce controlled mismatches that mimic anomalies by randomly replacing a patch in each test image with a corresponding patch from the training dataset at the same location, thereby generating ground-truth problematic regions. Knowing these regions allows us to quantitatively assess our method's precision in identifying artificially induced problematic regions. We conduct experiments on four datasets: C10, C100, SVHN, and PartImageNet. For the C10, C100, and SVHN datasets, we create ground-truth problematic regions by substituting a 10 by 10 patch in test images with a matching sample from the training data at the same location. For the high-resolution PartImageNet, we replace a 100 by 100 patch in each image. We then select 500 images from each dataset that exhibit the most significant increase in uncertainty after modification as our test subjects, indicating the presence of problematic regions. To identify these, we generate uncertainty maps for total uncertainty U_t that illustrate each pixel's contribution to it. We use these maps to locate the problematic regions by fitting a bounding box of the same size as the ground truth (10 by 10 for C10, C100, and SVHN; 100 by 100 for PartImageNet) around the area with the highest average attribution score, employing a straightforward brute-force search method to pinpoint the most likely location of the altered patch. Besides qualitative evaluation, our quantitative metrics include Intersection over Union (IoU) and anomaly detection accuracy (ADA). IoU calculates the overlap percentage between detected and actual anomalies divided by the union of these regions, while ADA measures the percentage of detections with an IoU above 0.5.

Qualitative Evaluation. The examples of uncertainty maps for all methods are shown in Figure 1. We observe that our method can clearly detect problematic regions that match the ground truth. It is worth noting that gradient-based methods, such as Grad, SmoothGrad, IG, and UA-Backprop, usually produce noisy "scattered" visualizations. Although other perturbation-based baselines may yield clear regions, they often lead to misleading results by pointing to inaccurate areas. Appendix B will provide additional examples across datasets, uncertainty maps for real images with high uncertainty, and some failure cases.

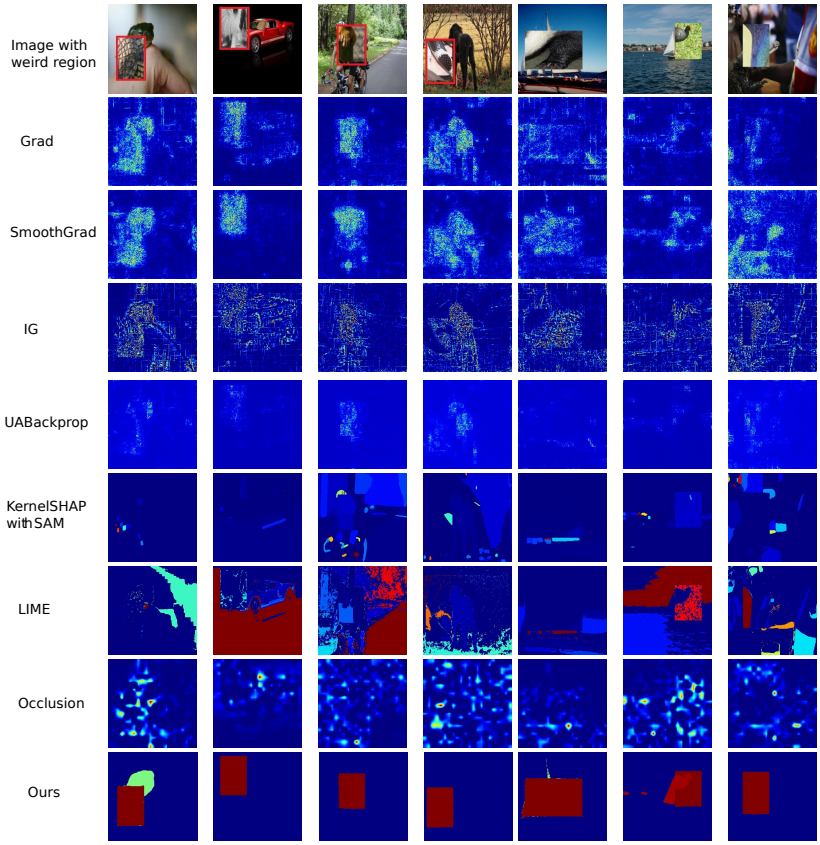

Fig. 1: Examples of uncertainty maps for identifying problematic regions on PartImageNet. Darker regions indicate a higher contribution to the uncertainty.

Quantitative Results. The experiment results are shown in Table 1. Our method significantly outperforms all baselines in detecting problematic regions across all datasets. For instance, on the SVHN dataset, our method achieves an IoU of 0.829 and an ADA of 0.830, marking a considerable improvement of approximately 50% in both IoU and ADA over LIME and Occlusion. It is worth noting that perturbation-based methods generally perform better than gradient-based methods in detecting problematic regions. When KernelSHAP is enhanced with SAM (KernelSHAP + SAM), it shows notable improvements, potentially because SAM provides valuable prior knowledge about potential problematic areas, offering a solid foundation for perturbation-based methods to identify candidates for perturbation. However, our method still surpasses KernelSHAP + SAM. If SAM is not available, our method can be applied at the pixel level, as discussed in Sec. 3.6. To avoid any confusion, results comparing our method with other baselines at the pixel level will be provided in Appendix C.1. Our method still consistently outperforms other baselines in the pixel-level setting.

Table 1: IoU ↑ and ADA ↑ for detection of problematic regions. Bold values indicate the best performance.

Method	C10		C100		SVHN		PartImageNet	
	IoU	ADA	IoU	ADA	IoU	ADA	IoU	ADA
Grad	0.109	0.054	0.282	0.250	0.288	0.180	0.147	0.065
SmoothGrad	0.297	0.226	0.322	0.132	0.312	0.184	0.161	0.075
IG	0.208	0.152	0.224	0.250	0.310	0.256	0.487	0.505
UA-Backprop	0.271	0.178	0.360	0.250	0.336	0.258	0.619	**0.695**
KernelSHAP	0.324	0.306	0.286	0.250	0.382	0.340	-	-
KernelSHAP + SAM	0.676	0.670	0.667	0.662	0.798	0.794	0.322	0.280
LIME	0.281	0.258	0.553	0.500	0.375	0.330	0.242	0.190
Occlusion	0.218	0.160	0.245	0.250	0.350	0.302	0.301	0.240
Ours	**0.713**	**0.700**	**0.750**	**0.750**	**0.829**	**0.830**	**0.633**	0.635

4.2 Faithfulness Test

Faithfulness in uncertainty attribution quantifies the accuracy with which a method's explanations reflect the actual influence of input features on the model's uncertainty. A method exhibits high faithfulness if the regions it identifies as significant truly correspond to areas in the input that, upon alteration, substantially affect the model's uncertainty. In this section, we will refine the most problematic pixels of the input, for example, by updating 2% of the pixels with the highest UA scores, and then observe the reduction in uncertainty after the alteration. To select the pixels with the highest UA scores, we compare all baselines at the pixel level. This is necessary because if one segment shares one score, the relative importance of each pixel remains unknown, preventing us from selecting the most problematic pixels.

The experiments are conducted on images with high uncertainty from the C10, C100, and SVHN testing datasets. Specifically, we select 500 samples with the highest total uncertainty from the testing data for each dataset. We exclude PartImageNet from the comparison because KernelSHAP and LIME are difficult to apply to high-resolution images at the pixel level. The linear regression used in these methods requires that the number of perturbations be at least larger than the number of pixels, which takes a significant amount of time and substantial memory. We consider two methods to refine the original inputs in their problematic regions. The first method utilizes gradient descent with uncertainty as the sole loss function. Specifically, we iteratively update the input $\mathbf{x}$ through $\mathbf{x} = \mathbf{x} + \eta \cdot \mathbf{M}(\mathbf{x}) \cdot \frac{\partial U(\mathbf{x})}{\partial \mathbf{x}}$ where η is the learning rate, $\mathbf{M}(\mathbf{x})$ is a binary map of the same size as x which highlights the pixels (i.e., 1 % of the total pixels) with largest UA score, and $U(\cdot)$ represents the function for estimating uncertainty. Through iteratively gradient descent, the uncertainty should be significantly reduced if $\mathbf{M}(\mathbf{x})$ shows the true problematic areas. The second method directly blurs the most problematic regions indicated by $\mathbf{M}(\mathbf{x})$, (i.e., 2 % of the total pixels) following the experiment settings of [39] for the blurring test. After alter-

Table 2: We evaluate attribution performance in terms of URR ↑ across four datasets, refining images by targeting the top 0.5% or 1% of pixels with high total uncertainty using iterative gradient descent, and the top 2% or 5% of pixels using blurring techniques. Bold values denote the best performance.

Method	Refining using Iterative Gradient Descent						
	C10		C100		SVHN		Avg. Performance
	0.5%	1%	0.5%	1%	0.5%	1%	0.5% + 1%
Grad	0.662	0.885	0.615	0.778	0.681	0.808	0.738
SmoothGrad	0.623	0.863	0.578	0.760	0.725	0.860	0.735
IG	0.621	0.865	0.581	0.761	0.700	0.838	0.728
UA-Backprop	0.706	0.903	0.261	0.436	0.714	0.853	0.646
KernelSHAP	0.731	0.904	0.635	0.779	0.733	0.838	0.770
LIME	0.787	0.927	0.458	0.573	0.256	0.411	0.569
Occlusion	0.822	**0.940**	0.297	0.453	0.420	0.592	0.587
Ours	**0.825**	0.924	**0.766**	**0.808**	**0.988**	**0.992**	**0.884**
Method	Refining using Gaussian Blurring						
	C10		C100		SVHN		Avg. Performance
	2%	5%	2%	5%	2%	5%	2% + 5%
Grad	0.343	0.440	0.099	0.136	0.053	0.031	0.184
SmoothGrad	0.331	0.418	0.079	0.126	0.109	0.105	0.195
IG	0.392	0.481	0.096	0.141	0.032	0.066	0.201
UA-Backprop	0.374	0.453	0.088	0.118	0.076	0.151	0.210
KernelSHAP	0.823	0.902	0.319	0.448	0.122	0.098	0.452
LIME	0.378	0.355	**0.415**	**0.547**	0.131	0.222	0.341
Occlusion	0.730	0.786	0.297	0.390	0.423	0.539	0.528
Ours	**0.860**	**0.918**	0.386	0.487	**0.869**	**0.939**	**0.743**

ing the original input $\mathbf{x}$ to get the updated input $\tilde{\mathbf{x}}$, we compare the uncertainty of them by calculating the Uncertainty Reduction Rate (URR): $1 - \frac{U(\tilde{\mathbf{x}})}{U(\mathbf{x})}$. For gradient descent refinement, we aim to update 0.5% and 1% of the total pixels, whereas for the blurring test, we refine 2% and 5% of the total pixels. More pixels are refined in the blurring test to achieve noticeable uncertainty reduction, as blurring is not as effective as directly optimizing uncertainty to attain the optimal value. The detailed settings will be shown in Appendix A.3.

The experiment results are shown in Table 2. our method stands out remarkably, achieving the highest average URR in both settings. Specifically, with an impressive average URR of 0.884 using iterative gradient descent, our method outperforms its closest competitor, KernelSHAP, by an impressive 11.4%. Similarly, in the blurring refinement, our method's average URR significantly surpasses that of Occlusion, the next best performer, by 21.5%. It is worth noting that perturbation-based LIME and Occlusion perform a little bit worse than gradient-based methods if the refinement is conducted using gradient descent. This may be because gradient-based methods provide the directions in terms of gradients for refining the images. If the refinement is conducted using blur-

Table 3: Problematic region detection results for the contribution of each component.

Method	C10		SVHN	
	IoU	ADA	IoU	ADA
Eq. (3)	0.333	0.294	0.371	0.325
Eq. (3) + SAM	0.648	0.638	0.709	0.712
Eq. (3) + SAM + LP	0.675	0.664	0.765	0.764
Eq. (3) + SAM + LP + Gumbel	**0.713**	**0.700**	**0.829**	**0.832**

ring, perturbation-based methods usually works better since they consider the blurring perturbation in the attribution process to estimate the contribution of each pixel. Overall, our method consistently outputforms other baselines, leveraging both advantangles of perturbation-based and gradient-based methods for enhanced faithfulness.

4.3 Ablation Studies

Contribution of Each Component. In this section, we aim to highlight the contribution of each component to the performance in detecting problematic regions. The results are presented in Table 3. The method, described as "Eq. (3)", solves Eq. (3) by relaxing $\mathbf{M}$ into a probability mask with a sigmoid function, $\mathbf{M} = \sigma(\mathbf{t}) \in [0,1]^{|\mathbf{M}|}$, where $\mathbf{t} \in \mathcal{R}^{|\mathbf{M}|}$ are the learnable parameters. The results indicate that all three components positively contribute to the improvement in detecting problematic regions. SAM contributes the most significantly, and Gumbel-sigmoid reparameterization is the second most significant contributor. More analysis will be shown in Appendix C.2.

Hyperparameter Sensitivity. Two hyperparameters, λ and τ, are essential for tuning our method. λ balances uncertainty loss with sparsity loss in Eq. (3), typically chosen to prioritize uncertainty loss over sparsity regularization. τ in Eq. (4) adjusts the approximation level for a binary mask, with a small τ (0.05 to 0.2) effectively achieving good binary approximations. Empirical results in Table 4 demonstrate that the performance is relatively stable across a range of these hyperparameters. Results for more datasets will be shown in Appendix C.3.

Table 4: Hyperparameter sensitivity for detecting problematic regions on C10 dataset.

Method	C10	
	IoU	ADA
$\lambda = 3, \sigma = 0.1$	0.713	0.700
$\lambda = 3, \sigma = 0.07$	0.679	0.686
$\lambda = 3, \sigma = 0.15$	0.688	0.682
$\lambda = 3, \sigma = 0.1$	0.713	0.700
$\lambda = 2.5, \sigma = 0.1$	0.677	0.680
$\lambda = 2, \sigma = 0.1$	0.682	0.680

Complexity Analysis. The empirical runtime for attributing one input is shown in Table 5. While our proposed methods can

achieve better UA performance, their runtime complexity depends on the number of gradient descent iterations needed for convergence when solving (5).

Typically, about 100-300 forward and backward passes are needed to generate one uncertainty map. Further analysis of the convergence of our method will be provided in Appendix C.4. Among the baselines, Grad is the most efficient, requiring only one forward and backward propagation. The time complexity of SmoothGrad and IG depends on the number of samples used for aggregating the gradients, which usually involves hundreds of samples. Although UA-Backprop theoretically only requires one forward and backward pass, in practice, it usually needs one pass per class. Compared to perturbation-based methods such as KernelSHAP and LIME, our method generally requires less time when applied to high-resolution images. KernelSHAP and LIME encounter challenges due to their linear regression component, which demands a sufficient number of perturbations and can be slow when dealing with a high variable dimension. It is worth noting that our method scales well with large images and can generate pixel-level UA maps for these images with complexity that increases linearly.

Table 5: Average runtime of attributing one input in seconds (s).

Method	C10 Runtime	PartImageNet Runtime
Grad	0.02s	5.94s
SmoothGrad	2.10s	33.3s
IG	2.14s	35.6s
UA-Backprop	0.26s	9.75s
KernelSHAP	1.63s	>200s
KernelSHAP + SAM	1.08s	113s
LIME	1.16s	>200s
Occlusion	0.15s	>200s
Ours	1.54s	103s

5 Conclusion

This research aims to develop a more faithful and explainable uncertainty attribution method for identifying problematic regions of the input. We formulate UA as an optimization problem to learn a binary mask that indicates problematic regions, thus enhancing our method's faithfulness by directly optimizing uncertainty as the objective. Three key enhancements are introduced. First, we propose SAM-based mask parameterization to generate more human-understandable uncertainty maps, potentially making the optimization process more efficient. Second, we introduce learnable perturbations to adaptively learn the perturbation parameters without the need for manual tuning. Third, we apply a probabilistic approach to learn the binary mask through Gumbel-sigmoid reparameterization under continuous optimization. For evaluation, we test our method's ability to detect synthetic problematic regions and estimate its faithfulness with real, problematic images, demonstrating our method's superior performance. We also discuss the limitations and future work of our method in Appendix D.

Acknowledgement. This work was supported in part by the National Science Foundation award IIS 2236026 and in part by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network.

References

1. Antorán, J., Bhatt, U., Adel, T., Weller, A., Hernández-Lobato, J.M.: Getting a clue: A method for explaining uncertainty estimates. arXiv preprint arXiv:2006.06848 (2020)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. Advances in neural information processing systems **30** (2017)
4. Depeweg, S., Hernandez-Lobato, J.M., Doshi-Velez, F., Udluft, S.: Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In: International Conference on Machine Learning. pp. 1184–1193. PMLR (2018)
5. Fong, R., Patrick, M., Vedaldi, A.: Understanding deep networks via extremal perturbations and smooth masks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2950–2958 (2019)
6. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE international conference on computer vision. pp. 3429–3437 (2017)
7. Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., Bloch, I.: Encoding the latent posterior of bayesian neural networks for uncertainty quantification. arXiv preprint arXiv:2012.02818 (2020)
8. Guo, H., Wang, H., Ji, Q.: Uncertainty-guided probabilistic transformer for complex action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20052–20061 (2022)
9. He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.N., Liu, S., Yang, C., Yuille, A.: Partimagenet: A large, high-quality dataset of parts. arXiv preprint arXiv:2112.00933 (2021)
10. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
11. Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., Bolukbasi, T.: Guided integrated gradients: An adaptive path method for removing noise. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5050–5058 (2021)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
14. Krizhevsky, A., Nair, V., Hinton, G.: The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html **55**(5) (2014)
15. Kwon, Y., Won, J.H., Kim, B.J., Paik, M.C.: Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. Computational Statistics & Data Analysis **142**, 106816 (2020)

16. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles (2016), http://arxiv.org/abs/1612.01474
17. Ley, D., Bhatt, U., Weller, A.: {\delta}-clue: Diverse sets of explanations for uncertainty estimates. arXiv preprint arXiv:2104.06323 (2021)
18. Ley, D., Bhatt, U., Weller, A.: Diverse, global and amortised counterfactual explanations for uncertainty estimates. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 7390–7398 (2022)
19. Louizos, C., Welling, M.: Multiplicative normalizing flows for variational bayesian neural networks. In: International Conference on Machine Learning. pp. 2218–2227. PMLR (2017)
20. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888 (2018)
21. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
22. Maddox, W.J., Izmailov, P., Garipov, T., Vetrov, D.P., Wilson, A.G.: A simple baseline for bayesian uncertainty in deep learning. In: Wallach, H., Larochelle, H., Beygelzimer, A., e-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 13132–13143. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/9472-a-simple-baseline-for-bayesian-uncertainty-in-deep-learning.pdf
23. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern recognition **65**, 211–222 (2017)
24. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
25. Ng, I., Zhu, S., Fang, Z., Li, H., Chen, Z., Wang, J.: Masked gradient-based causal structure learning. In: Proceedings of the 2022 SIAM International Conference on Data Mining (SDM). pp. 424–432. SIAM (2022)
26. Perez, I., Skalski, P., Barns-Graham, A., Wong, J., Sutton, D.: Attribution of predictive uncertainties in classification models. In: The 38th Conference on Uncertainty in Artificial Intelligence (2022)
27. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421 (2018)
28. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
29. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
30. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International conference on machine learning. pp. 3145–3153. PMLR (2017)
31. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
32. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
33. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)

34. Srinivas, S., Fleuret, F.: Full-gradient representation for neural network visualization. Advances in neural information processing systems **32** (2019)
35. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
36. Tierney, L.: Markov chains for exploring posterior distributions. the Annals of Statistics pp. 1701–1728 (1994)
37. Valdenegro-Toro, M.: Deep sub-ensembles for fast uncertainty estimation in image classification. arXiv preprint arXiv:1910.08168 (2019)
38. Wang, H., Ji, Q.: Diversity-enhanced probabilistic ensemble for uncertainty estimation. In: Uncertainty in Artificial Intelligence. pp. 2214–2225. PMLR (2023)
39. Wang, H., Joshi, D., Wang, S., Ji, Q.: Gradient-based uncertainty attribution for explainable bayesian deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12044–12053 (2023)
40. Wang, H., Wang, S., Ji, Q.: Semantic attribution for explainable uncertainty quantification. In: International Workshop on Epistemic Uncertainty in Artificial Intelligence. pp. 101–112. Springer (2023)
41. Watson, D., O'Hara, J., Tax, N., Mudd, R., Guy, I.: Explaining predictive uncertainty with information theoretic shapley values. Advances in Neural Information Processing Systems **36** (2024)
42. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 681–688 (2011)
43. Wen, Y., Tran, D., Ba, J.: Batchensemble: an alternative approach to efficient ensemble and lifelong learning. arXiv preprint arXiv:2002.06715 (2020)
44. Xu, S., Venugopalan, S., Sundararajan, M.: Attribution in scale and space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9680–9689 (2020)
45. Yang, Q., Zhu, X., Fwu, J.K., Ye, Y., You, G., Zhu, Y.: Mfpp: Morphological fragmental perturbation pyramid for black-box model explanations. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 1376–1383. IEEE (2021)
46. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
47. Zhang, R., Li, C., Zhang, J., Chen, C., Wilson, A.G.: Cyclical stochastic gradient mcmc for bayesian deep learning. arXiv preprint arXiv:1902.03932 (2019)
48. Zhang, Y., Wang, H., Kephart, J.O., Ji, Q.: Body knowledge and uncertainty modeling for monocular 3d human body reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9020–9032 (2023)